A Self-Aware Machine Platform in Manufacturing Shop Floor Utilizing MTConnect Data

Linxia Liao[1], Raj Minhas[2], Arvind Rangarajan[3], Tolga Kurtoglu[4], Johan de Kleer[5]

[1,2,4,5] *Palo Alto Research Center, Palo Alto, CA, 94304, USA*
linxia.liao@parc.com
raj.minhas@parc.com
tolga.kurtoglu@parc.com
dekleer@parc.com

[3] *General Electric Global Research, San Ramon, CA, 94583, USA*
arvind.rangarajan@ge.com

ABSTRACT

We propose a framework of self-aware machines based on data collected using the MTConnect protocol. Beyond existing applications of OEE (Overall Equipment Effectiveness) reporting, the proposed framework integrates multiple sources of information for work-piece and machine condition monitoring, and equipment time to failure prediction in manufacturing processes, and provides feedback to shop supervisor. Firstly, we propose a method to predict component wear and failure based on operational data. ICP (Interactive Closest Point) algorithm is used to find the best matching tool path given a certain tool number to identify similar machining processes. The result of ICP tool path matching, together with other parameters such as spindle speed, feed rate and tool number, are used to adaptively cluster the machining processes. For each process cluster, a particle filter based prognostic algorithm is used to predict tool wear and/or spindle bearing failure. Secondly, we propose to use anomaly detection methods to detect changes in normal behavior of the machines. Various machine learning algorithms are utilized to detect anomalies based on real-time data, and a voting mechanism is used to decide when to trigger an alarm. Thirdly, the axes traverse is aggregated to provide a measure of the wear on various axes in the machine, which is correlated to errors in position comparing to the commanded positions and nominal tool paths. Spindle load verse rotating speed is also examined to facilitate shop floor scheduling to avoid damage caused by unintentionally excessive machine usage. The proposed framework has been demonstrated using published data

from two Mazak machine tools.

1. INTRODUCTION

Sparked by IT megatrends, manufacturers are currently undergoing an operational transformation with increased agility and efficiency. Key technologies influencing this change include digital manufacturing, cloud computing, mobile application, and big data. At the intersection of these technologies there is an opportunity to create a self-aware machine platform in manufacturing shop floor. With the advancement of sensing technology and automation, more information can be derived to facilitate better collaboration and decision making.

Some of the most critical factors, influencing the output of a machining process, are related to tooling, operating parameters, and the ability of a machine tool to maintain its accuracy and repeatability. Changes due to wear or failure of critical machine tool components can lead to significant losses in production and unexpected downtime. One of the current barriers of condition monitoring systems is that the collected sensor data are not well correlated with the in-process machining operating conditions, which compromises the prediction accuracy. Another barrier is that the typical assumptions underlying the prediction of time to failure algorithms (e.g. exponential fault growth) are rarely applicable in real machining. In addition, existing systems operate independently, and impose proprietary interfaces and machine communication protocols that can lead to excessive time consuming and expensive installations.

The goal of the proposed framework is to develop a self-aware system capable of integrating multiple sources of information for work-piece and machine condition monitoring, and equipment time to failure prediction in manufacturing

processes. Currently, the primary applications developed using MTConnect
(MTConnect, 2009) data are focused on the visualization and reporting of OEE (Overall Equipment Effectiveness) and history of alarms. The proposed method goes beyond reporting to provide insight for cell operators on accumulated damage and use automatic clustering for process grouping with particle filter based prognostics using time series data to provide early warning systems for tool wear. Rigid body registration algorithms are used to automatically identify segments of tool paths that can be used to predict or reinforce tool wear prediction. Multiple anomaly detection algorithms with a voting mechanism are used to detect process anomalies across machines. We believe that machine self-awareness will drive the value chain from traditional fail-and-fix, preventive maintenance, condition based monitoring towards self-adaptive, self-analyzing and coordinated assets (see Figure 1)

2. THE PROPOSED FRAMEWORK

The proposed framework uses MTConnect data alone to derive information of health condition estimation and prediction for machine components, process anomalies detection across machines using machine learning methods, provide shop floor planning recommendation using statistics.

2.1. Data Collection and Preprocessing

For demonstrating our framework, we use data provided at a public URL for the MTConnect challenge. A query post (e.g. `http://66.42.196.109:5605/sample?count=2000`) is sent periodically to the MTConnect enabled machine IP address. The query returns an XML (Extensible Markup Language) formatted file which contains all the data published from the machine. Since we query periodically, the data returned by a query may contain some data that was also returned as part of a previous query. To avoid data redundancy, we check the sequence numbers returned from the query result to record data when it is updated. Using the tags 'nextSequence', 'firstSequence', and 'lastSequence', we ensure that 'nextSequence' is greater than 'lastSequence' and 'nextSequence' increases by the count number compared to its previous value (e.g. count number is set to 2000 in the query example shown above). A snapshot of the data XML file is shown in Figure 2.

A parser is written to obtain the time stamps and values of the variables from the tags in the returned data file. The variables that we obtained include x-axis position, y-axis position, z-axis position, spindle load, x-axis load, y-axis load, z-axis load, feed rate, feed rate override, spindle speed, spindle speed override, and tool number. The data is updated when the value of a variable is changed. Hence, for a certain time stamp, there may be no value for a variable because it is not updated at the time stamp. If there is no value available, the

Figure 1. A vision of self-aware machine.

Figure 2. An example of MTConnect data file in XML format.

previous value is inserted at the time stamp since the value hasn't changed yet. After the parsing and insertion, a vector of a time stamp and the values of all the aforementioned variables are obtained. This allows us to get a matrix of data indexed by multiple time stamps.

2.2. Component Level Health Monitoring and Prediction

One of the characteristics of a self-aware machine is to be able to detect its components degradation and predict future failure. The components (e.g. spindle, cutting tool, and feed axis) on a machine are often used under different machining processes in a manufacturing shop floor. A machining process in our research is defined as a cutting tool with the same tool number sharing similar tool paths with the same non-zero spindle speed and feed rate (overridden value) associated with a certain time period. For each process, the spindle power data were recorded as wear indicators. An adaptive clustering method is applied to cluster the different processes. Prediction is made using a filtering method to predict component failures with data from the specific process as well as data from other processes using the same tool. The prediction provides insight into every single process, which not only guides the maintenance decision makers to take proactive actions on the machine component to avoid unplanned
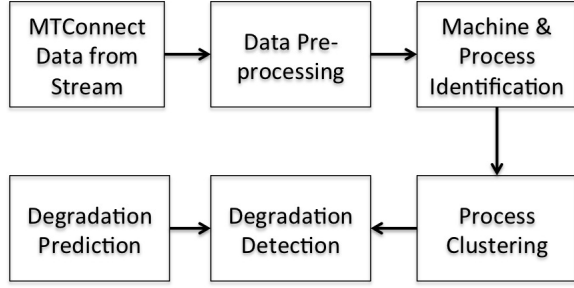
590

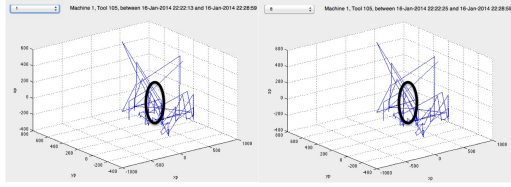Figure 3. Flowchart of component level health monitoring and prediction.

Figure 4. The tool paths of similar machining processes.

downtime, but also assists the process planners to track the production drawbacks to improve their process design. The flowchart is shown in Figure 3.

- Machine and Process Identification
  Different machines are using different IP addresses to publish the data. The identification of the machine will be determined by the IP address used in the query post described in Section 2.1. For a specified cutting tool, the tool path consists of multiple x, y, and z positions. The spindle speed and feed rate change during machining. For the same part, x, y, and z positions determine the shape of the tool path in 3-D space (shape space). The spindle speed, feed rate and time form another 3-D space (parameter space). For two machining processes, if the same cutting tool is used for the entire machining process and the shape space and the parameter space are both matching, we assume these two machining processes are similar processes. The shape spaces of two similar processes are shown in Figure 4. There are small variations in the circled area. This could be happening because the MTConnect protocol has a limitation in the sampling rate. Other than that, the entire tool paths of these two processes are very similar.
  We use ICP (Interactive Closest Point) algorithm (Savoye, 2012) to determine how the shape space and parameter space match. ICP is a commonly used algorithm to align two free-form point clouds in 3-D space. It optimizes the transformation matrices such as scaling, rotation, and translation applied on the target shape to min-

imize the error with the source shape. It has been successfully used in many fields such as manufacturing (3-D surface inspection), and healthcare (medical image segmentation). We use ICP algorithm to find the best matching machining processes. Let us denote the original 3-D space points cloud as $source$, the transformed points cloud as $tranform$, and the targeted points cloud as $target$. The operation matrix of rotation, scaling and translation are T, b and c, respectively. After the operation we obtain

$$transform = b * source * T + c \qquad (1)$$

The ICP algorithm optimizes the operation matrix of T, b and c so that the difference (denoted as d) between $tranform$ and $target$ is minimized. The difference shows the extent to which $source$ and $target$ are different. The smaller the difference, the better the match/overlap between $source$ and $target$. The difference between the shape spaces is denoted as d_s, and the difference between the parameter space is denoted as d_p. The matching measure is denoted as $d_a = [d_s, d_p]$.

- Process Clustering
  Machines are usually programmed to perform different jobs under various machining processes depending on the tasks. To compare the condition of the machine, we need to group the similar processes into a cluster with in which the analysis is performed to derive the health condition. The data stream may contain a brand new process that has not been experienced before. An adaptive clustering method is used to automatically cluster the machining processes into different clusters. If a new machining process is detected (i.e. it does not belong to any existing process clusters), a new process cluster is assigned. If a machining process belongs to an existing cluster, the process is assigned to that cluster and the centroid of the cluster is updated. To determine whether a process belongs to an existing cluster or not, a T2 limit is applied on the matching measure d_a. Let the mean value of the matching measure of an existing cluster be $\hat{d}_a$ and the covariance be s. The T2 statistics for the matching measure of a process is calculated by

$$T2 = (d_a - \hat{d}_a) * s^{-1} * (d_a - \hat{d}_a)' \qquad (2)$$

The T2 control limit is calculated by

$$T2_{limit} = \frac{(N-1)(N+1)p}{N(N-p)} F_\alpha(p, N-p) \qquad (3)$$

where $F_\alpha(p, N-p)$ is the $100\alpha\%$ confidence level of F-distribution with p and $N-p$ degrees of freedom. If the $T2$ statistic is below the $T2_{limit}$, the process belongs to an existing process cluster; otherwise a new cluster is created for the process.
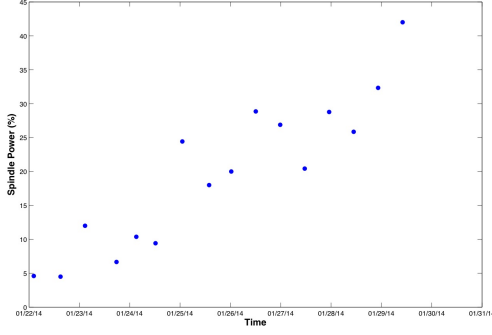
Figure 5. Degradation of cutting tool No. 63.

- Degradation Detection

  After similar processes are grouped into clusters, we can perform degradation detection within each cluster. We assume that the spindle power increase is proportional to the increased severity of tool wear for similar machining processes. The local trend of the power increase may vary (e.g. there may be stochastic variations locally). However, the overall trend of the power should be increasing over time. Hence, a monotonicity criterion is used to detect the increasing trending of the spindle power. Monotonicity is defined in (Coble & Hines, 2009) as:

$$Monotonicity(F) = \frac{\#d/dF > 0}{n - 1} - \frac{\#d/dF < 0}{n - 1} \quad (4)$$

  where F is the measurement, n is the number of measurement in a period of time. F represents a feature and d/dF is the derivative. The maximum value of $Monotonicity$ equals to 1 only if the feature is monotonically increasing. The value of monotonicity indicates the increasing trend of the spindle power, which indirectly indicates the degradation of the cutting tool. Figure 5 shows the detected trend of the cutting tool number 63.

  This analysis will be performed within all the process clusters. If multiple processes belong to a same cutting tool and degradation trend has been detected with these processes, it is more certain that the cutting tool is wearing.

- Degradation Prediction

  If a degradation trend is detected, we can extrapolate the trend to infer the remaining cuts under the same process given a preset threshold of the power. A particle filter (Chen, Zhang, Vachtsevanos, & Orchard, 2011) can be adapted for the prediction due to its capabilities to cope with system non-linearity and estimate prediction uncertainty. The prediction is made using a continuous Bayesian update method assuming the fault growth following a physics-based system degradation model (e.g. the Paris' Law), which is widely used as the fatigue crack growth model. The system degradation was assumed to

be a first-order Markov process, i.e. the current state was only dependent upon the last state. In this case, we observed that the degradation trend was closely following a second order polynomial model such as:

$$X_k = a_k t_k + b_k t_k^2 + c_k \quad (5)$$

where X_k is the system state (tool wear in this case), t_k is the time at step k, and a_k, b_k, c_k are the parameters of the second order polynomial model. We can write Eq.(5) into the format of a Markov model as follows:

$$
\begin{aligned}
X_k &= a_k t_k + b_k t_k^2 + c_k \\
&= a_k(t_{k-1} + \Delta t) + b_k(t_{k-1} + \Delta t)^2 + c_k \\
&= a_k t_{k-1} + b_k t_{k-1}^2 + c_k \\
&\quad + a_k \Delta t + 2b_k t_{k-1} \Delta t + b_k \Delta t^2 \\
&= X_{k-1} + (a_k + 2b_k t_{k-1})\Delta t + b_k \Delta t^2 \quad (6)
\end{aligned}
$$

The parameter identification and state estimation can be performed in parallel. The prediction (median of the particles) of the remaining cuts for the degradation situation shown in Figure 5 is 13 give 70% of spindle power as the threshold. This information can alert the maintenance team to change the cutting tool before it fails.

2.3. Process Anomaly Detection Across Machines

Anomaly detection (Barnett & Lewis, 1994), (Hodge & Austin, 2004) is an important concept for a self-aware system. An anomaly is simply an exception or deviation from the typical usage (tools, power, speed etc.) and does not necessarily imply a malfunction. For example, machining a new part or using a new tool or working with a new type of material may all be deviations from the previous usage of a machine. However, these are intended (and desired) deviations - on the other hand, if the power usage is unusually high despite unchanged job parameters then it may point to an underlying condition. So a self-aware machine can indicate to the operator that it is experiencing a significant deviation from its typical behavior - the operator can decide whether the deviation is a cause for concern. In fact, the operator can annotate the behavior for future use. So if the anomaly is just a desired new behavior then it can be labeled as such and the machine will know not to flag it in the future. On the other hand, if it is an indication of an underlying condition then it can be labeled with the diagnosis and the machine can flag it appropriately in the future. In this section, we show how anomaly detection can be performed on MTConnect data to identify deviations in usage. While not as informative as the approaches mentioned in Section , anomaly detection can be very scalable as it need not rely on models of failure.

As mentioned in Section 2.3, we analyze data from an MTConnect stream. Let us look at a snippet of this data shown in Table 1. The first six columns provide a time stamp for the

data while the remaining columns provide details about the job (tool ID, feed rate, spindle speed, tool path, and spindle power) - we use the job parameters for our analysis. In the literature, there are a number of popular approaches to anomaly detection. Here, we consider three: 1) self organizing maps (SOMs), 2) regression, and 3) Mahalanobis distance.

2.3.1. Self Organizing Maps (SOMs)

SOMs (Kohonen, 2001) are a natural way to organize an incoming stream of data into a grid of cells - a (typically Euclidean) distance metric is used to assign new data instances to cells containing similar data. As data accumulates, some cells will become very dense and will represent the typical behavior/usage of the machine. If a new data instance is assigned to sparsely populated cell then that would indicate a deviation from the typical behavior/usage. If this behavior is desirable or intended then the cell can be labeled as such. Otherwise, it can indicate undesired behavior or malfunction. For this data, a SOM is shown in Figure 6. While the data is high-dimensional, for ease of visualization we have only shown spindle speed (x-axis) and spindle power (y-axis). We start with a 7x7 grid evenly distributed on the space spanned by the expected range of the variables. Then we assign points to the cells in an incremental manner based on the Euclidean distance. After a data point has been assigned, the cells are warped to have a greater resolution in areas of high density (i.e. areas representing usual behavior) - please see (Rougier, Boniface, & Universit, 2011) for more details. The gray lines in Figure 6 represent the Voronoi partition (`http://en.wikipedia.org/wiki/Voronoi_diagram`) of this grid where each partition represents the extent of the corresponding node - a data point within a partition is assigned to the node associated with it. Due to the warping, the structure of the data clearly stands out. The lower left corner has small and dense cells representing the typical usage of the machine. The space of large spindle speeds and power is very sparse. There is a clear anomaly in the top right corner corresponding to spindle power of 87 units and spindle speed of 3127 rpm - in addition, there are many sparse cells corresponding to higher than usual values of speed and power. If a new data point falls in a sparse or hitherto unseen region, it can be flagged for review. The operator can choose to investigate and annotate the cell for future reference.

2.3.2. Multivariate Regression

Another way to look at this problem of self-awareness is from the perspective of relationships between the variables. In a control system such as a CNC machine, the high level requirements (e.g. the tool path) are translated into low level specifications (e.g. feed rate, spindle speed etc.) which are then met using control inputs (e.g. spindle power). So it may be quite normal for power usage to be high if the required speed is high. If we can learn the normal relationship
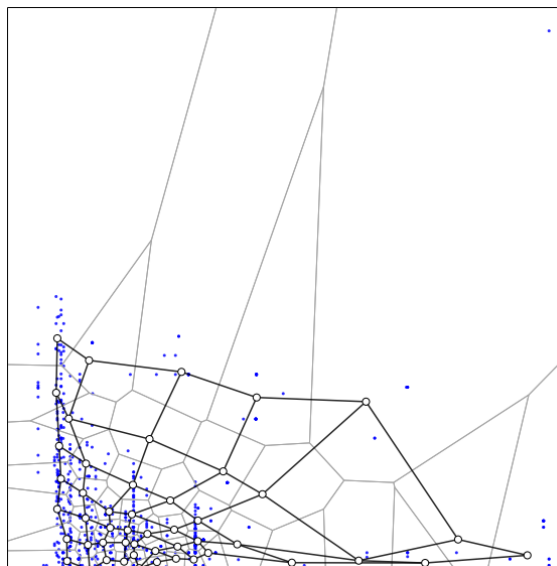

Figure 6. A Self-Organizing Map for MTConnect Data from a Mazak Machine

Table 2. Processed MTConnect Data

tool ID	dur-ation	spindle speed	feed rate	dist-ance	spindle power
0	0.083	400	1.19	0.81	13
0	0.70	1131	26.84	194.82	7
...	...	...	...	...	...

between the different variables then it should be possible to raise a flag when the variables of a new data instance exhibit a significantly different relationship. In this section, we show how multivariate regression may be used to learn the relationship between variables.

Before performing regression, we need to pre-process the data. In Section 2.3, we mentioned that ICP path matching as a approach for analyzing the tool path - it ensures that the analysis performed is invariant with respect to affine transformations of the tool path. The primitive for our regression analysis is not the entire tool path but rather the sampling interval of the data collection process - executing the entire tool path may take many minutes but the data being analyzed is sampled every few seconds. So rather than analyzing the entire tool path, we analyze the distance traveled by the tool during a sampling instance. This is just a design choice - domain expertise can be used to pick a different primitive. After pre-processing, we get data of the following form:

Here *tool ID* is a categorical variable[1] while the others are real numbers - we try to learn a model to predict spindle power based on the other variables. There are many modeling ap-

---

[1]There are 36 distinct tool IDs: 0, 10, 102, 104, 107, 108, 109, 111, 112, 115, 117, 118, 120, 17, 2, 20, 24, 25, 3, 32, 4, 44, 45, 5, 52, 58, 63, 65, 69, 70, 74, 77, 88, 90, 92, 98

Table 1. MTConnect Data

year	month	day	hour	minute	second	tool ID	feed rate	spindle speed	x	y	z	spindle power
2014	1	23	14	51	28	0	1.19	400	2.11	-32.46	-70	13
2014	1	23	14	51	33	0	1.19	400	0	-32.46	-69.14	13
...	...	...	...	...	...	...	...	...	...	...	...	...

proaches for regression but we are specifically interested in two characteristics: 1) ability to provide a prediction interval for new data points, and 2) ability to build accurate models without making assumptions about the nature of relationship between the variables. The first requirement (prediction interval estimation) is necessary for defining anomalies (deviations) in a structured manner but the second requirement (assumption-free modeling) is just a convenience to enable automation. There are many options but quantile regression forests (Meinshausen, 2006) are ideally suited for this scenario and that is what we used for this analysis. They provide a reasonable fit to the data and give us the ability to estimate prediction intervals based on user defined quantiles. Let Q_α be defined as

$$Q_\alpha(x) = \inf \{P(Y \leq y | X = x) \geq \alpha\} \qquad (7)$$

Then Q_α represents the $\alpha-$quantile for the conditional distribution of a variable Y conditioned on a vector variable X. If Y is the variable being predicted (spindle power in our example) then Q_α defines its $\alpha-$quantile conditioned on the prediction variables X (tool ID, duration, spindle speed, feed rate, and distance in our example). For this analysis, we use $[Q_{0.025}, Q_{0.975}]$ as the prediction interval and designate a new data instance as anomalous if the actual spindle power lies outside the prediction interval. Compared to the SOM approach, this approach has the advantage that we explicitly model the relationship between spindle power (dependent variable) and the other variables (independent variables). The notion of prediciton interval is also a big advantage as it provides a systematic approach to detecting outliers. The prediciton interval will be small if we have a high confidence in our prediction so even small unexpected deviations outside the prediction interval may be flagged. On the other hand, it has the disadvantage that we can only flag anomalies in the value of the independent variable conditioned on the independent variables - we cannot flag anomalies in the independent variables themselves (since they are considered inputs into the model). Typically, excessive deviations in the control signal are good indicators of underlying conditions so this is not a big drawback.

For this dataset, the quantile regression forest achieves reasonable accuracy in predicting the spindle power ($R^2 = 0.74$). However, we are not interested in the actual predictions per se but rather in large errors in those predictions (i.e. values that lie outside $[Q_{0.025}, Q_{0.975}]$). The graph in Figure 7 shows
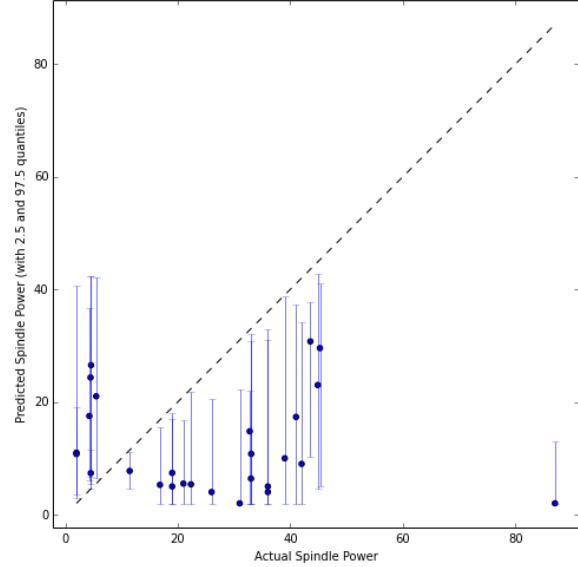

Figure 7. Outlier Detection using Quantile Regression Forest

such deviations. As in the case of SOMs, the instance where the spindle power is 87 stands out as a clear outlier. Most of the other outliers are cases where the actual value lies just outside the prediction interval.

2.3.3. Robust Mahalanobis Distance

If the data are assumed to be samples from a multivariate normal distribution then Mahalanobis distance can be used to detect outliers. In that case, outliers are data points that are samples from a different distribution rather than extreme values of the multivariate normal distribution. This has the advantage that we don't need to choose a cutoff point for labeling a point as outlier - we simply look for points that likely came from a different distribution (see (Filzmoser, Garrett, & Reimann, 2005) for more details). Of course, the normality assumption may not be satisfied in reality - in fact, it is not satisfied for the data set being used here. In that case, we can still use Mahalanobis distance to look for outliers without relying on distributional assumptions. One approach is to transform the data into the principal component space and look for the outliers in the space spanned by the top few principal components. Since principal components are aligned with directions of maximal variance, that makes it easier to spot the outliers. Also, by looking in the reduced space of the top principal components, it increases the signal to noise ratio. Using ap-
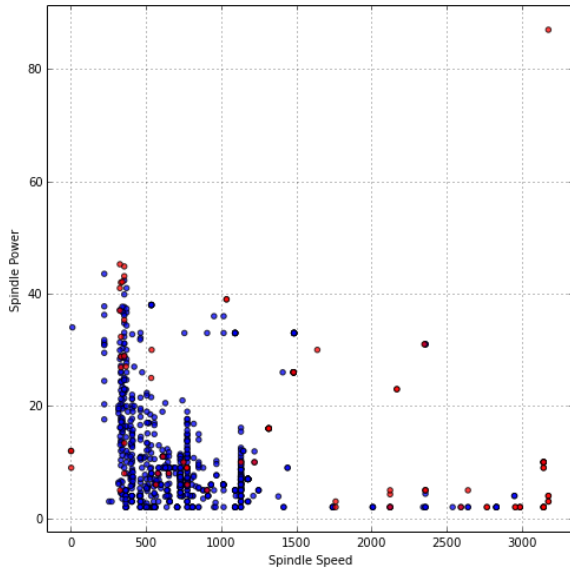
Figure 8. Mahalanobis Distance Based Outlier Detection

propriate normalization (see (Filzmoser, Maronna, & Werner, 2008) for more details), the Euclidean distance in the principal component space is equivalent to Mahalanobis distance in the original space. In the absence of any distributional assumptions, (Filzmoser et al., 2008) proposes a measure of *outlyingness* of a data instance based on its Mahalanobis distance. We use that same measure in our analysis here.

The results are show in Figure 8 - the outliers are shown in red[2]. The instance where spindle power is 87 is again identified as a clear outlier in addition to some others.

2.3.4. Ensemble of Outlier Detection Methods

In this section, we discussed three outlier detection approaches, namely, self-organizing maps, multivariate regression, and robust Mahalanobis distance. There are many other other methods that could be applied. All these methods make different assumptions and have different strengths and weaknesses. We can combine them into an ensemble that can raise flags based on some predetermined policy. For example, if the cost of failure is very high then the ensemble may flag a data instance as an outlier if any member of the ensemble determines the data instance to be an outlier (this would be an *OR* policy). Alternatively, if the cost of disruption of workflow outweighs the cost of failure then the ensemble may flag a data instance as an outlier only if all members of the ensemble agree (this would be an *AND* policy). In most scenarios, a good policy might be for the ensemble to flag a data instance as an outlier if a large fraction of the ensemble members agree (this would be a *MAJORITY* policy).

---

[2]This multivariate analysis included duration, feed rate, spindle speed, distance, and spindle power but we only show the spindle speed and power in the graph for ease of visualization.

2.4. Shop Floor Planning Recommendation

Another aspect of machine self-awareness is that the machines are able to compare their usage and performance with each other. The information can be fed back to the shop floor planning trying to avoid damage due to unintentionally excessive usage by rescheduling the machining tasks.

The spindle data can be used to estimate spindle damage as the bearing life is proportional to $load^3 * rpm$ (revolutions per minute). The aggregate axes traverse provides a measure of the wear on various axes in the machine (an estimate of the way damage). This can be correlated to error in position if either commanded position is available via MTConnect protocol or nominal tool paths are available to switch the axis to condition based maintenance. This recommendation provides insights by shop defined rules ifor switching parts between machines if any axis travels beyond a threshold greater than twice that of a comparable machine in the same time frame.

Figure 9 contains an overview about a cell of machines. The machines are identified by the individual MTConnect Stream. We use the data from two machine provided by MTConnect challenge (`http://66.42.196.109:5605/current` and `http://66.42.196.109:5606/current`). The figure has three distinct sets of information presented: recommendations for the cell based on data, histogram plot of spindle rpm (revolution per minute) weighted by the load at the specific rpm, and total traverse compared across different feed axes on the machine. MTConnect provides insight into usage of machines both absolute and relative to each other in a cell when aggregated over time. The histogram of the spindle loads weighted by the time spent at various spindle speeds provide a relative estimate of remaining useful life (RUL) of the spindle bearings. This information can be fed back to the scheduling systems depending on the shop's maintenance policy. For example, if all machines will be taken down around the same time for service, this can be used to balance the spindle loads across machine. Similar analysis can be employed to balance travel of various drive axes by shifting parts appropriately. These include rotating the fixtures based on current state and scheduled tool paths.

This helps shop supervisors balance usage across machines at a deeper level than utilization to reduce excessive damage accumulation on a single machine in a cell while reducing unexpected downtime for individual machines. The recommendation will enable manufacturing shops to move from scheduled maintenance to condition based maintenance based on true damage accumulation.

3. CONCLUSION AND DISCUSSION

The framework we have developed is scalable with broad applicability for milling, drilling, turning machines in various configurations. It can be configured from cell level to
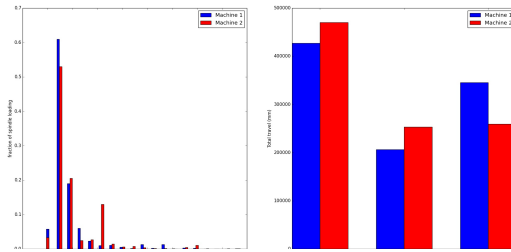
Figure 9. Shop floor recommendation for spindle and axis planning.

plant level with minimal effort and is applicable for small and medium-sized or large enterprises. It also has broad based applicability for various industries including fabricating industrial components, such as automotive engine, medical device, or aerospace parts. Only part of the MTConnect data is considered in our research. More variables can be used to obtain the machine health information from a broader view. The sampling rate has certain limitations as mentioned in the previous section. More information can be derived by combining operational data with external sensor data (e.g. vibration, acoustics signal) to gain more insight about the machine component health, e.g. (Liao & Pavel, 2012) and (Liao, Edmondson, & Ludwig, 2012).

Machine self-awareness could shift the industry from a reliance on a preventative paradigm (checking performance and replacing parts on a set schedule, regardless of whether there is an immediate need for these activities), to a predictive paradigm (schedule maintenance before failure actually happens). Self-aware machines will positively impact production time, cost, and quality of any manufacturing plant by reducing unplanned downtimes, adapting for work-piece variability, and enabling specification of fault-tolerant process plans.

REFERENCES

Barnett, V., & Lewis, T. (Eds.). (1994). *Outliers in statistical data*. Wiley New York.

Chen, C., Zhang, B., Vachtsevanos, G., & Orchard, M. (2011). Machine condition prediction based on adaptive neuro–fuzzy and high-order particle filtering. *IEEE Transactions on Industrial Electronics*, *58*(9), 4353-4364.

Coble, J., & Hines, J. W. (2009). Identifying optimal prognostic parameters from data: a genetic algorithms approach. In *Proceedings of the annual conference of the prognostics and health management society.*

Filzmoser, P., Garrett, R. G., & Reimann, C. (2005). Multivariate outlier detection in exploration geochemistry. *Computers & Geosciences*, *31*(5), 579-587.

Filzmoser, P., Maronna, R., & Werner, M. (2008). Outlier identification in high dimensions. *Computational Statistics & Data Analysis*, *52*(3), 1694-1711.

Hodge, V. J., & Austin, J. (2004). A survey of outlier detection methodologies. *Artificial Intelligence Review*, *22*(2), 85-126.

Kohonen, T. (Ed.). (2001). *Self-organizing maps*. Springer.

Liao, L., Edmondson, Z., & Ludwig, H. (2012). Plug and prognose - condition monitoring, diagnosis and life time prediction. *ATP edition*, *54*(10), 52-56.

Liao, L., & Pavel, R. (2012). Machine tool feed axis health monitoring using plug-and-prognose technology. In *Proceedings of the 2012 conference of the society for machinery failure prevention technology.*

Meinshausen, N. (2006). Quantile regression forests. *Journal of Machine Learning Research*, *7*, 983-999.

MTConnect. (2009). *Mtconnect standard part 1-overview and protocol, version 1.01* (Tech. Rep.). MTConnect Institute.

Rougier, N., Boniface, Y., & Universit, L. (2011). Dynamic self-organising map. *Neurocomputing*, *11*(74), 1840-1847.

Savoye, Y. (2012). Iterative cage-based registration for dynamic shape capture. In *Acm siggraph 2012 posters.*

BIOGRAPHIES

Linxia Liao Linxia Liao's research interests include predictive and prescriptive analytics; system and components fault diagnostics and prognostics; signal processing; and machine learning algorithms as well as their integration on embedded systems. Currently, he is developing device fleet health management solutions for intelligent transportation systems. Prior to joining PARC, Linxia worked as a research scientist with Siemens Corporation, Corporate Technology (previous Siemens Corporate Research) located in Princeton, NJ. He conducted research and implemented various prognostics and health management-related applications in the fields of manufacturing, energy, and transportation. He also previously worked at Siemens Technology-To-Business (TTB) Center in Berkeley, CA to transfer the patented 'Methods for prognosing mechanical systems' technology from the university to industry applications. Linxia received his Ph.D. degree in Industrial Engineering from the University of Cincinnati, where he conducted research at the NSF I/UCR Center for Intelligent Maintenance Systems (IMS). Linxia has one issued patent and seven pending patents, and he has published one book chapter and 20+ papers in leading journals and conferences.

Raj Minhas Raj Minhas is the Program Manager for Prognostics and Health Management and is responsible for the strategy and execution for the commercialization of the related technologies. Prior to PARC, he was the Director of Xerox Research Center India where he led its growth, development, and outreach for two years. He held a variety of leadership roles at Xerox Research Center Webster includ-